REMARKS

Status of this application

Claims 1-16 are pending.

In the Office Action mailed on December 29, 2006, Claims 1-16 were rejected under 35 U.S.C. 102(e) as being anticipated by Sridhar U.S. Application Publication No. 2003/0221162 (hereinafter "Sridhar"). Claims 1-16 were further rejected under 35 U.S.C. 102(e) as being anticipated by Koskas Patent 6,711,563 (hereinafter "Koskas").

This amendment makes clarifying amendments to claims 1 and 11-14 to more clearly define the invention.

Reconsideration of the rejections of the claims based on the teachings of Sridhar and Koskas is requested for the reasons detailed below. The remarks that follow discuss the individual rejections in the order they were presented in the outstanding action.

The rejections based on Sridhar

Claims 1-16 were rejected under as being anticipated by Sridhar. Reconsideration is requested since Sridhar does not disclose or suggest the subject matter set forth in the rejected claims.

Claim 1 is directed to a relational database management system (RDBMS) for storing and analyzing network data stored in two relational tables (a generic node table and a generic link table). The data in these two tables define a set of nodes, and links between nodes, forming a network wherein each of nodes represents an object of interest, and each of the links represents a relationship between two of the nodes. Claim 1 expressly defines the content of both the generic node table and the generic link table. Further, claim 1 further requires the presence of an application program interface that enables external executing application programs to perform operations including (1) creating the generic node table and the generic link table, (2) storing data describing network nodes in the rows of the generic node table, (3) storing data describing links between these nodes in the generic link table, and (4) performing standard operations on the data in the generic node and link tables.

The Examiner suggests that Sridhar's relational database management system stores and analyzes "network data" and cites numerous paragraphs of Sridhar. In those paragraphs, Sridhar uses the terms "node" and "link" to describe how graph theory is used to model the relationships between data tables defined by a schema. As Sridhar succinctly explains in 0077, data tables are represented by "nodes" and "links" are employed to model the primary key / foreign key relationships between the records of one table and records of its related tables. Sridhar further explains how links may be contained in the rows of link tables, each link table row containing a pair of keys which associate a record (row) in one table with a record (row) in another table. For example, Fig. 9 shows a patient table 902 and physician table 904 (each table being a "node") and shows three linking tables 906, 908 and 910, each of which contains rows, with each row in a linking table containing patient-id and a physician-id keys that establish a relationship between particular patient (described by a row in the patient table 902) and a particular physician (described by a row in the physician table 904). The link table 906 contains rows which hold keys that relate a patient and a referring physician, the link table 908 contains rows which hold keys that associate a patient with a primary care physician, and the link table 910 contains rows which hold keys that associate a patient with a secondary physician.

It is understood to be the Examiner's position that one of Sridhar's tables such as tables 902 and 904 in Fig. 9 or tables 1012 and 1014 seen in Fig. 10) constitutes a "generic node table" as defined by claim 1, and that Sridhar's linking tables (such as the tables 906, 908 and 910 in Fig. 9 or the table 1000 in Fig. 10) corresponds to the generic link table of claim 1.

While Sridhar's tables (which are there called "nodes") such as the patient table 902, the physicians table 904, the suppliers table 1012, contain table rows which contain data describing object of interest (patients, physicians, suppliers and parts), the objects described by the rows of these tables are not the nodes of a data network joined by links as defined by claim 1. Sridhar's linking tables define relationships between rows of different tables. Sridhar's data tables themselves are the "nodes" and Sridhar's linking tables contain relationships between tables (nodes). In contrast, in applicant's invention as set forth in claim 1, there is a single generic node table that contains rows the describe the set of nodes forming the network, and a single generic link table that contains rows that describe relationships (links) between the nodes (described by the rows of the single generic node table). As amended, claim 1 requires the presence of "a generic link table containing a plurality of link table rows each of which contains data describing a link between two nodes in said set of nodes and further identifying the two node table rows in said generic node table that contain data describing each of said two nodes." Sridhar's link tables don't contain links between rows of the single generic row table, and do not contain additional data that describes a link.

Moreover, claim 1 states requires the presence of "an <u>application program interface</u> which enables executing application programs to create said generic node table and said generic link table, to store data describing nodes in said generic node table, to store data describing links between said nodes in said generic link table, and to perform a plurality of standard operations on the data in said generic node table and said generic link table." Sridhar nowhere suggests the creation of an application program interface that permits external executing application programs to create tables, store data in tables, or perform standard operations on data in tables. Indeed, Sridhar does not disclose or mention an application program interface for any purpose.

Instead, Sridhar provides an interactive system that allows the user (a website developer) to use a webpage interface to define a data model which the system converts into group of data views, and a webpage user interface that allows the user to choose the data views that are to be presented as web pages. Sridhar does not provide an API that permits an external executing application to create and load tables and to thereafter perform standard operations on the data in those tables as required by claim 1. Sridhar is instead a single program that interfaces directly with the web site developer (user) and not with external executing application programs as require by claim 1.

The Examiner suggests that Sridhar describes an application program interface which enables executing application programs to create a node and link tables at paragraphs 0021, 0056, 0081, 0091, 0164. All of these paragraphs describe how the Sridhar system interfaces with a human user to simplify and automate the task of building web pages that display data contained in an RDBMS, but none suggests that an API is made available to an external executing application program to permit it to create and/or perform standard operations on table data representing a network.

The Examiner further suggests that Sridhar discloses an API for storing data describing nodes in a node table in paragraphs 0078, 0121, 0127, 0131, 0133. The cited passages describe how the Sridhar system processes data definitions provided by the user using a web interface to create data views that are displayed in web pages. These paragraphs do not suggest that Sridhar's system provides an API which enables external executing application programs to store data in network data tables as required by claim 1.

The Examiner further suggests that Sridhar discloses an API for storing data describing links between said nodes in a link table at paragraph 0078. That paragraph describes the internal makeup of Sridhar's data structures, but does not suggest that an API is provided for storing data in link tables as claimed.

The Examiner suggests that Sridhar describes an API the enables an application perform a plurality of standard operations on the data in said node table and said link table at paragraphs 0008, 0009, 0048, 0057. Paragraphs 0008 and 0009 describe the shortcomings of prior art development tools for building web pages, paragraph 0048 describes how the Sridhar system builds webpage bases systems that are platform independent, and paragraph 0057 describes how the system creates back-end Java code to implement the system which the website designer wants to build and install. None of these paragraphs suggest the creation of an API that enables external executing application programs to performing network data analysis.

Claim 2 states that the network is a "logical network." Applicants' specification, at [041] defines a "logical network" as follows: "A logical network contains connectivity information but no geometric information. This is the model used for network analysis. A logical network can be treated as a directed graph or undirected graph, depending on the application." Sridhar's system for building websites is not designed to, and is not disclosed as being capable of, defining a "logical network" as claimed using RDBMS tables.

The Examiner states that a "web page is inherently logical network" citing the abstract and paragraphs 0021 and 0079. None of these paragraphs describe a logical network that contains connectivity information of the type set forth in claim 1 as further defined by claim 2.

Claim 3 requires that the rows of both the generic node table and the generic link table include a cost attribute. Paragraph [044] of applicants' specification defines "cost" as being a non-negative numeric attribute that can be associated with links or nodes for computing such things as the minimum cost path (the path that has the minimum total cost from a start node to an end node)."

The Examiner suggests that "Sridhar discloses that each of said node table rows contains data specifying a node cost attribute (weight) associated with said given node (See paragraphs 0051, 0077) and wherein each of said link table rows further contains a link cost attribute (weight) associated with a link (See paragraphs 0009, 0077, 0094, 0104)."

At 0077, Sridhar notes that "links may be weighted or non-weighted" but does not describe how weighted links are used, and there is no suggestion that nodes are weighted. The reference in 0051 to "weight" refers to the weight of a part, not the "cost" of a node. None of the passages cited by the Examiner discloses the a cost attribute as claimed placed in any table of any kind.

<u>Claims 4 and 5</u> are dependent on claim 3 and further require that one of the standard operations that an application program can perform using the RDBMS API is to identify a particular path having one or more stated characteristics, specifically the "least cost" in claim 5. Applicants' specification at [039] defines a "path" as "an alternating sequence of nodes and links, beginning and ending with nodes, and typically with no nodes and links appearing more than once."

The Examiner states that "Sridhar discloses that the standard operations include at least one path identification procedure for analyzing the said network data to identify a particular path having one or more stated cost characteristics, citing paragraphs 0127, 0132 and 0009, 0077, 0094, 0104. Paragraphs 0127, 0132 and 0094 describe how Sridhar's system traverses a UDM tree structure to ascertain (dereference) a data value at a leaf node. This process is not a standard operation performable by API and does not identify a path having stated cost characteristics as claimed. Paragraph 0077 says links may be weighted, but Sridhar does not describe any use of such weighed links. Paragraphs 0009 and 0104 do not appear to be relevant.

<u>Claim 6</u> requires that one of the standard operations is the identification of a path consisting of an alternating sequence of nodes and links having defined characteristics. The Examiner suggests that Sridhar discloses that said standard operations include analyzing said network data to identify a path consisting of an alternating sequence of nodes and links having defined characteristics, citing paragraphs 0127-0132. While Sridhar does describe traversing paths between rows of tables using the link pointers in those rows, that traversal is performed internally within the Sridhar system to search for referenced data items, and is not a

standard operation that an external executing application program invokes via an API as claimed.

Claim 7 requires that the RDBMS of claim 1 further includes a path table containing a plurality of path table rows each of which contains data describing a path consisting of an alternating sequence of nodes and links. An example is seen in applicant's Fig. 4.

The Examiner states that "Sridhar discloses that said system further includes a path table containing a plurality of path table rows each of which contains data describing a path consisting of an alternating sequence of nodes and links (See paragraphs 0127- 0132)." The Examiner does not identify any particular row of any particular table in Sridhar that can be said to describe an "alternating sequence of nodes and links." Sridhar's link tables (e.g. tables 906, 908 and 910 seen in Fig. 9) contain rows each containing a primary key and a foreign key and can be said to define a single link between two nodes (i.e. data table rows). Consequently, for the purposes of this amendment, applicants rely on limitations expressed in parent claim 1 for the allowability of claim 7.

Claim 8 is dependent on claim 7 and further requires that an RDBMS API standard operation is used to identify a path and place information describing that path in one row of the path table as defined in parent claim 7.

The Examiner suggests that such an operation is disclosed in paragraphs 0127-0132, but a careful review of those paragraphs reveals no such operation. Reconsideration or clarification of this rejection is requested.

Claim 9 is also dependent on claim 7 and further requires the creation of a path-link table containing one ordered set of path-link table rows associated with each given path described in said path table, each of said path table rows containing information identifying one link in the sequence of links in said given path.

The Examiner asserts that Sridhar discloses a path-link table having the content claimed in paragraphs 0126-0134 and/or 0148. No such teaching has been found in the cited paragraphs and reconsideration (or clarification of the rejection is requested.

Claim 10 is dependent on claim 9 and requires that an RDBMS API standard operation is used to identify a path and place information describing that path in one of the path table rows and also placing a sequence of links in a path link table. The Examiner rejected claim 10 based on the paragraphs 0127- 0132 without further explanation. A careful review of those paragraphs reveals no such operation. Reconsideration or clarification of this rejection is requested.

Claim 11 states the standard RDBMS API operations includes loading node and link data into the node and link tables respectively from a database.

The Examiner suggests that "Sridhar discloses that said standard operations include loading (retrieval, extract data) node and link data into said node and link tables respectively from a database," citing paragraphs 0051, 0059, 0045, 0047, 0082. None of these paragraphs discuss loading data into tables from a database, and Sridhar does not disclose API operations all, let alone the specific API operations that load the generic node table and the generic link table from a database.

Claims 12, 13 and 14 are directed to the use of the invention to define and analyze a "spatial network." As defined in applicants' specification at [042], a "spatial network" contains both connectivity information and geometric information. The manner in which applicants' RDBMS network model stores geometric information is described in detail beginning at [090] in applicant's specification. The Examiner suggests that "Sridhar discloses that said network is a spatial network and wherein each of said node table rows includes a column (See Figures 9, 10) for storing the identification of a geometry object (See Figures 1, 2), which specifies the shape and location of one of said nodes" and "specifies the geometry of one of said links." Claims 12-14 have been amended by this response to more clearly define this aspect of applicants' invention. Nothing in Figs. 1, 2, 9 or 10 describes a network that is a spacial network or a node table that includes rows that identify a geometry object that specifies an ordered sequence of vertices that are connected by straight line segments or circular arcs that define the shape and location of a node as set forth in claim 12 as amended, or identifies a geometry object that specifies the geometry of one of said links as specified in claim 13. Reconsideration is requested.

Note that, while Sridhar's table 1014 seen in Fig. 10 contains rows that identify parts (shoe polish, tooth paste, paper clips), these objects are not objects that specify anything.

Claim 15 requires that each of said node table rows further contains a level column for holding a hierarchy level. The Examiner states that Sridhar discloses that each of said node table rows further contains a level column for holding a hierarchy level, citing paragraphs 0068, 0113, 0174, 0177, 0178 and Figure 13. The cited paragraphs discuss the fact that the relationship between Sridhar's tables may be represented as a hierarchy of nodes, but none of the cited paragraphs discloses or suggests that a table could or should include a level column for holding a hierarchy level.

Claim 16 is dependent on claim 15 and further requires that each of said node table row further contains a parent column for holding the identification of a parent node within the hierarchy established by said level column. The Examiner Sridhar discloses that each of said node table rows further contains a parent column for holding the identification of a parent node within the hierarchy established by said level column, citing paragraphs 0068, 0113, 0174, 0177, 0178 and Figure 13. None of those paragraphs, or Fig. 13, discloses a table row that identifies a parent node within a hierarchy, and more specifically not a hierarchy that is established by the values in a level column as specified in parent claim 16.

Claims 1-16 are accordingly believed to be patentable over Sridhar for the reasons given above.

The rejections based on Koskas

Claims 1-16 were rejected under 35 U.S.C. 102(e) as being anticipated by Koskas Patent 6,711,563 ("Koskas"). Contrary to the assertions made by the Examiner, Koskas does not disclose the subject matter set forth in claims 1-16 for the reasons explained below.

Claim 1 is directed to a relational database management system (RDBMS) for storing and analyzing network data stored in two relational tables (a generic node table and a generic link table). The data in these two tables define a set of nodes, and links between nodes, forming a network wherein each of nodes represents an object of interest, and each of the links represents a relationship between two of the nodes. Claim 1 expressly defines the content of both the generic node table and the generic link table. Further, claim 1 requires the presence of an application program interface that enables external executing application programs to perform operations including (1) creating the generic node table and the generic link table, (2) storing data describing network nodes in the rows of the generic node table, (3) storing data describing links between these nodes in the generic link table, and (4) performing standard operations on the data in the generic link and node tables.

Koskas describes a method of organizing a database management system to achieve efficient query processing by building a virtual data graph (VDG) structure comprising a virtual flat file and sorted thesaurus files pointing to rows of the virtual flat file. The virtual flat file combines associated records from tables that are linked together.

As explained by Koskas, in a relational database, linked tables are employed to eliminate redundancies to achieve more efficient memory usage. Two tables are linked if one of them (the source table) has a link column containing a foreign key designating a record in another table (the target table). Linked tables can be represented in an acyclic directed graph structure such as a hierarchical tree in which a root table is only a source table and not a target table, leaf nodes which are only target tables and not source tables, and mid-level tables which are both. (see Koskas at col. 7, line 40 et seq.). Linked tables can be combined to form a flat file table structure in which each row contains columns containing the data from linked rows of related tables. The flat file structure can alternatively be reduced to a linked table that contains nothing but links pointing to the data in the related tables. Koskas further creates inverted index structures called "word thesauruses" which contain data values (words) and occurrence data which identify the rows of the flat file that contain those data values. These thesauruses may then be employed to more rapidly identify and process desired data by avoiding the need to conduct row-by-row scans for data of interest.

The Examiner suggests that "Koskas discloses a generic node table containing a plurality of node table rows each of which contains data describing a given node in said network," citing col. 25, lines 60 to col. 26, line 14 and claims 77, 79. The cited passages describe how the "Virtual Data Graph" index structure is modified when new data is added but it is not clear which of the various tables discussed in that passage are contended by the Examiner to meet the definition of a "generic node table" in claim 1.

The Examiner further asserts that a generic link table containing a plurality of link table rows each of which contains data describing a link between two nodes in said network and identifying each of said two nodes is described by Koskas at col. 26, lines 16-22 and lines 43-56, Figure 9. That passage, and Fig. 9, describes the effect of changing a link in a source data table on the entries in the link table (that represents the flat file); accordingly, it is understood that the

Examiner is contending that a link table of the kind shown in Fig. 9 corresponds to the generic link table as set forth in claim 1.

Claim 1 as amended requires the presence of "a generic link table containing a plurality of link table rows each of which contains data describing a link between two nodes in said set of nodes and further identifying the two node table rows in said generic node table that contain data describing each of said two nodes." The rows in Koskas' link table do not describe a link between two nodes (rows) in a single generic node table. Instead, Koskas' link table rows contains links between rows of different tables, not links between rows of a single table, because the purpose of Koskas' link table is to establish relationships between tables, not within the rows of a single table. Moreover, Koskas' link table rows contain a collection of links (keys to foreign tables), but do not contain "data describing a link ... and further identifying the two node table rows in said generic node table that..." Claim 1 requires that the link table entries both identify two node table rows that are linked and contain "data describing the link." Koskas' link table rows do not have such content.

Moreover, claim 1 states requires the presence of "an <u>application program interface</u> which enables external executing application programs to create said generic node table and said generic link table, to store data describing nodes in said generic node table, to store data describing links between said nodes in said generic link table, and to perform a plurality of standard operations on the data in said generic node table and said generic link table." Koskas nowhere suggests the creation of an application program interface that permits executing application programs to create node tables and link tables, store data in such tables, or perform standard operations on data in tables. Koskas' linking tables and other components of the Virtual Data Graphs are used internally within the database engine to more rapidly perform searching and processing, and their creation and use occurs internally. No mechanism is suggested by which the RDBMS user, or a separate external executing application program, can create, or store data in, or perform standard operations on the data in, these internal data structures. The cited passages at col. 3, lines 1-27, Figures 8-10, and col. 4, lines 1-58 have been reviewed but found not to contain any disclosure or suggestion of an API as set forth in claim 1.

Accordingly, reconsideration of the rejection of claim 1 and its dependent claims based on Koskas is requested.

Claim 2 states that the network is "logical network." Applicants' specification, at [041] defines a "logical network" as follows: "A logical network" contains connectivity information but no geometric information. This is the model used for network analysis. A logical network can be treated as a directed graph or undirected graph, depending on the application." The Examiner suggests that Koskas discloses that said network is a logical network in Figures 4-7. Those figures show examples of the linked relationship between RDBMS tables that contain information about insurance company clients, policies, accidents and brokers. This data does not describe a logical network composed of nodes and links. The data is organized in linked tables, but it does not "describe" the nodes and links that form a logical network, and the specific generic node table and generic link table structures defined by parent claim 1 are not used.

Claim 3 requires that the rows of both the generic node table and the generic link table include a cost attribute. The description of the node record seen in Fig. 6 does not include a node cost attribute. The Examiner states that "Koskas discloses that each of said node table rows contains data specifying a node cost attribute associated with said given node and wherein each of said link table rows further contains a link cost attribute associated with a link, citing col. 51, lines 51-59; Col. 52, lines 43-53. These paragraphs discuss the cost of RAM circuits needed to implement the Kostas system, and the cost of increasing the capacity of the system. That, of course, has nothing to do with placing data specifying a node cost attribute and a link cost attribute in the node table rows and link table rows respectively. Kostas does not teach that.

Claim 4 is dependent on claim 3 and further requires that one of the standard operations that an application program can perform using the RDBMS API is to identify a particular path having one or more stated characteristics. The Examiner states that "Koskas discloses that said standard operations include at least one path identification procedure for analyzing the said network data to identify a particular path having one or more stated cost characteristics, citing col. 8, lines 21-33, col. 51, lines 51-59; col. 52, lines 43-53. The passage in column 8 discusses the path that exists between root and leaf nodes in Kostas' linked tables, not a path in a network described by data, and col. 8 says nothing about standard API operations, or operations for identifying particular paths have stated cost characteristics. The passages in col. 51 and 52, noted above with respect to claim 3, are irrelevant.

__Claim 5__ is also dependent on claim 3 and further specifically requires that one of the standard operations is the identification of the least cost path from a stated start to end node in the network. The Examiner states that "Koskas discloses that said standard operations include at least minimum cost path identification procedure for analyzing the said network data to identify the path that has the minimum total cost from a stated start node to a stated end node," again citing col. 51, lines 51-59 and col. 52, lines 43-53. As noted above, these passages have nothing to do with using cost attributes to identify least cost paths in a network.

__Claim 6__ requires that one of the standard operations performed by the RDBMS API is the identification of a path consisting of an alternating sequence of nodes and links having defined characteristics. The Examiner states that "Koskas discloses that said standard operations include analyzing said network data to identify a path consisting of an alternating sequence of nodes and links having defined characteristics, citing col. 8, lines 63 to col. 9, line 11. The cited passage discusses scanning data in tables from a leaf table to a root table in order to find particular data values or to insert null records in linking tables. This is not a standard operation performed by the RDMBS API nor does it identify paths having defined characteristics. Reconsideration is requested.

__Claim 7__ requires that the RDBMS of claim further includes a path table containing a plurality of path table rows each of which contains data describing a path consisting of an alternating sequence of nodes and links. An example is seen in applicant's Fig. 4. The Examiner states that "Koskas discloses that said system further includes a path table containing a plurality of path table rows each of which contains data describing a path consisting of an alternating sequence of nodes and links, citing Figures 1-8. But none of the tables shown in Figs. 1-8 contain path table rows __each of which__ contains data describing a sequence of nodes and links as claimed. The source tables seen in Figs. 2 and 3 contain single links which point to rows in target tables, Figs. 4-7 are graphs showing table relationships, and the flat file table in Fig. 8 has no links at all. There is no table row shown anywhere in these figures that contains data describing a sequence of nodes and links.

__Claim 8__ is dependent on claim 7 and requires that an RDBMS API standard operation is used to identify a path and place information describing that path in one row of the path table. The Examiner states that " Koskas discloses that said standard operations include at least one path identification procedure for analyzing said network data to identify a particular path having stated characteristics and for placing information describing said particular path in one of said path table rows," citing col. 3, lines 54-67; col. 4, lines 1-58; col. 8, lines 63 to col. 9, line 11 and Figures 4-7. None of these passages describes a standard operation (or any other operation) for identifying a path having a stated characteristic and then placing information describing the identified path in one of the path table rows as defined in parent claim 7.

<u>**Claim 9**</u> is also dependent on claim 7 and requires the creation of a path-link table containing one ordered set of path-link table rows associated with each given path described in said path table, each of said path table rows containing information identifying one link in the sequence of links in said given path. The Examiner cites col. 8, lines 63 to Col. 9, line 11 in support of his conclusion that "Koskas discloses that said system further includes a path- link table containing one ordered set of path-link table rows associated with each given path described in said path table, each of said path table rows containing information identifying one link in the sequence of links in said given path." As noted above with respect to parent claim 7, Koskas does not describe a path table as claimed, and further does not describe a path-link table that contains rows describing each link in the sequence of links in a given path. Kostas describes nothing of the sort in the cited passage.

<u>**Claim 10**</u> is dependent on claim 9 and requires that an RDBMS API standard operation is used to identify a path and place information describing that path in one of the path table rows and also placing a sequence of links in a path link table. The Examiner states that "Koskas discloses that said standard operations include at least one path identification procedure for analyzing said network data to identify a particular path having stated characteristics and for placing information describing said particular path in one of said path table rows and for placing information describing the sequence of links in said particular path in said path-link table" citing col. 3, lines 54-67 and col. 4, lines 1-58 (Kostas SUMMARY OF THE INVENTION). Nothing in that SUMMARY describes standard API operations, identifying paths having stated characteristics, a path table of the type claimed, a path-link table of the type claimed, or a mechanism for loading those tables with data about an identified path.

<u>**Claim 11**</u> states the standard RDBMS API operations include loading node and link data into the generic node table and the generic link table respectively from a database The Examiner

states that "Koskas discloses that said standard operations include loading node and link data into said node and link tables respectively from a database," citing col. 23, lines 43-58. The cited passage describes a procedure shown in Fig. 24 for updating a thesaurus table in the Virtual Data Graph and not API operations for loading node and link data into generic node and link tables of the type claimed in parent claim 1.

Claims 12, 13 and 14 are directed to the use of the invention to define and analyze a "spatial network." As defined in applicants' specification at [042], a "spatial network" contains both connectivity information and geometric information. The manner in which applicants' RDBMS network model stores geometric information is described in detail beginning at [090] in applicant's specification. The Examiner states that "Koskas discloses that said network is a spatial network and wherein each of said node table rows includes a column for storing the identification of a geometry object, which specifies the shape and location of one of said nodes, citing col. 50, lines 41-58, col. 51, lines 4-18; and col. 4, lines 10-23. But nothing in these cited passages describes a network that is a spacial network or a node table that includes rows that identify a geometry object that specifies an ordered sequence of vertices that are connected by straight line segments or circular arcs that define the shape and location of a node as set forth in claim 12 as amended, or identifies a geometry object that specifies the geometry of one of said links as specified in claim 13. Reconsideration is requested.

Claims 15 and 16 requires that each of said node table rows further contains a level column for holding a hierarchy level and that each of said node table row further contains a parent column for holding the identification of a parent node within the hierarchy established by said level column. The Examiner states, "Regarding Claim 15, Koskas discloses that each of said node table rows further contains a level column for holding a hierarchy level (See Col. 7, lines 14-67, Col. 20, lines 41-59)," and "Regarding Claim 16, Koskas discloses that each of said node table rows further contains a parent column for holding the identification of a parent node within the hierarchy established by said level column (See Col. 53, lines 40-50)." But a review of the cited passages in columns 7, 20 and 53 fails to reveal any disclosure of suggestion of a table that contain a level column for holding a hierarchy level (claim 15) or of parent column for holding the identification of a parent column within a hierarchy established by a level column (claim 16)

Claims 1-16 are accordingly believed to be patentable over Koskas for the reasons given above.

Conclusion

It is submitted that neither Sridhar nor Koskas disclose applicants' invention as claimed. Both of these references teach relational database systems, but do not describe the specific generic node table and generic link tables for implementing a general purpose framework for storing and analyzing network data as set forth in independent claim 1, nor do they describe or suggest the additional data structures and operations described in the dependent claims.

This application as now presented is accordingly believed to be in condition for allowance.

Respectfully submitted,

Dated: October 27, 2007

Charles G. Call, Reg. 20,406

Certificate of Transmission under 37 CFR 1.8

I hereby certify that this *Amendment* is being transmitted by facsimile to the central facsimile number of the U.S. Patent and Trademark Office, (571) 273-8300, on October 27, 2007.

Dated: October 27, 2007

Signature ________________
Charles G. Call, Reg. No. 20,406
USPTO Customer No. 021253
215 West Huron, Unit 2
Chicago, IL 60610-3331
Ph. (312-265-0900
call@patentsoft.com